

Amendments to the Claims:

Please replace all prior versions, and listings of claims in the application with the following listing of claims.

Listing of claims

Claim 1 (currently amended): A deinterleaving method for ~~processing data, comprising:~~

sorting a sequence of N data items from a first order to a second order~~[[;]], wherein N is an even number, the sequence comprises data items relating to two different sets of data items, and the first order is an interleaving of the two different sets of data items such that the data items from the two different sets of data items are arranged in a list or array alternating one sample relating to each of the two different sets of data items, the method comprising:~~

~~withdrawing at least a first data item~~ a pair of data items from said sequence, the pair of data items comprising a first data item and a second data item having ~~[[a]]~~ respective first and second ~~position~~ positions ~~[[from]]~~ within said sequence;

~~for each of said withdrawn data items, determining a destination position for said withdrawn data item~~ within said sequence; and

~~for each of said determined destination positions, determining whether said determined destination position contains any data item, if so replacing the data item of said determined destination position with [[the]] a respective one of the withdrawn data [[item]] items, otherwise inserting a respective one of the first data item and the second data item at said determined destination position,~~

~~wherein the sequence is indexed from 0 to $N-1$, and wherein the destination position of any incorrectly positioned data item is:~~

~~the index of an original position of the incorrectly positioned data item divided by 2, if the index of the original position of the incorrectly positioned data item is even; or~~

~~N divided by 2 and added to the index minus 1 of the original position of the incorrectly positioned data item divided by 2, if the index of the original position of the incorrectly positioned data item is odd.~~

Claim 2 (currently amended): The deinterleaving method according to claim 1, wherein each of the destination ~~position~~ positions is calculated based on the index of a respective one of the first ~~position~~ and second positions and the number of data items of said sequence.

Claim 3 (previously amended): The deinterleaving method according to claim 1, wherein two data items are repositioned in each sequence of steps.

Claim 4 (previously amended): The deinterleaving method according to claim 1, wherein the method is an in-place method carried out within a memory having a set of memory locations.

Claim 5 (previously amended): The method according to claim 1, wherein said sequence comprises an even number of data items, and wherein the number of data items relating to a first set of data items of said sequence is equal to the number of data items relating to a second set of data items of said sequence.

Claim 6 (previously amended): The deinterleaving method according to claim 1, wherein data items relating to a first and a second set of data items, respectively, are arranged alternating in said sequence before sorted, and wherein the data items when sorted within said sequence are grouped into consecutive data items having consecutive positions.

Claim 7 (currently amended): The deinterleaving method according to claim 5, wherein the ~~[[two]]~~ first pair of data items to be repositioned in ~~[[the]]~~ a same repositioning sequence are selected as one data item relating to each of the ~~first and second~~ two different sets of data items, and wherein said ~~[[two]]~~ first pair of data items ~~[[are]]~~ is selected as any ~~other~~ data items other than the first and last data items of the sequence.

Claim 8 (currently amended): The deinterleaving method according to, claim 5 wherein the ~~[[two]]~~ first pair of data items to be repositioned are selected as the data items stored at the center positions of said sequence.

Claim 9 (currently amended): The deinterleaving method according to claim 1, further comprising the steps of:

if the destination ~~position contains~~ positions contain no data ~~[[item]] items~~,
determining ~~at least one~~ two incorrectly positioned data ~~[[item]] items~~ to reposition; and
repositioning said ~~at least one~~ two incorrectly positioned data ~~[[item]] items~~.

Claim 10 (currently amended): The deinterleaving method according to claim 9, wherein for each of the two incorrectly positioned data items to be repositioned, the position of the ~~at least one~~ incorrectly positioned data item to be repositioned is determined as:

the position preceding a first destination position, which did not contain any data item; and/or

the position following a second destination position, which did not contain any data item.

Claim 11 (canceled)

Claim 12 (currently amended): A deinterleaving device for sorting a sequence of N data items from a first order to a second order, wherein N is an even number, the sequence comprises data items relating to two different sets of data items, and the first order is an interleaving of the two different sets of data items such that the data items from the two different sets of data items are arranged in a list or array alternating one sample relating to each of the two different sets of data items, the deinterleaving device comprising:

~~a memory having a set of memory locations for storing the sequence of data items;~~

a processor for sorting the data items; and

a buffer for storing ~~at least a first data item~~ the sequence of data items at ~~a first~~ respective memory location buffer locations within the buffer;

wherein said processor is adapted to withdraw ~~said data item~~ a pair of data items from said sequence from said buffer, the pair of data items comprising a first data item and a second data item having respective first and second buffer locations within said buffer, and said processor is further adapted to determine, for each of said withdrawn data items, a destination memory buffer location for said withdrawn data item, and, for each of said determined destination buffer locations, to determine whether said determined destination memory buffer location contains any data item, if so replacing the data item of said determined destination memory buffer location with [[the]] a respective one of the withdrawn

data ~~[[item]]~~ items, otherwise inserting a respective one of the first data item and the second data item at said determined destination ~~memory~~ buffer location,

wherein the buffer locations are indexed from 0 to $N-1$, and wherein the destination buffer location of any incorrectly positioned data item is:

the index of an original buffer location of the incorrectly positioned data item divided by 2, if the index of the original buffer location of the incorrectly positioned data item is even; or

N divided by 2 and added to the index minus 1 of the original buffer location of the incorrectly positioned data item divided by 2, if the index of the original buffer location of the incorrectly positioned data item is odd.

Claim 13 (currently amended): The deinterleaving device according to claim 12, wherein the processor is adapted to calculate each of the destination ~~position~~ positions based on the index of a respective one of the first ~~position~~ and second buffer positions and the number of data items of said sequence.

Claim 14 (previously amended): The deinterleaving device according to claim 12, wherein the processor is adapted to reposition two data items in each repositioning sequence.

Claim 15 (previously amended): The deinterleaving device according to claim 12, wherein the processor comprises a register file, and the repositioning of data items is done in-place in said memory.

Claim 16 (currently amended): The deinterleaving device according to claim 12, wherein said ~~memory~~ buffer comprises an even number of memory locations.

Claim 17 (currently amended): The deinterleaving device according to claim 14, wherein the processor is adapted to select the ~~[[two]]~~ first pair of data items to be repositioned in ~~[[the]]~~ a same repositioning sequence as one data item relating to each of ~~a first and a second set~~ the two different sets of data items, and select said ~~[[two]]~~ first pair of data items as any ~~other~~ data item other than the first and the last data items of the sequence.

Claim 18 (currently amended): The deinterleaving device according to claim 16, wherein the processor is adapted to select the ~~[[two]]~~ first pair of data items to be repositioned as the data items stored at the center memory locations of the memory.

Claim 19 (currently amended): The deinterleaving device according to claim 12, further adapted to:

if the destination ~~memory location contains~~ buffer locations contain no data ~~[[item]]~~ items, determine whether all data items of said sequence are positioned at their correct ~~memory~~ buffer location;

if any data item is stored at an incorrect ~~memory-buffer~~ location, determine ~~at least one~~ two incorrectly positioned data ~~[[item]]~~ items to reposition; and

reposition said ~~at least one~~ two incorrectly ~~stored~~ positioned data ~~[[item]]~~ items.

Claim 20 (currently amended): The deinterleaving device according to claim 19, wherein for each of the two incorrectly positioned data items to be repositioned, the ~~memory~~ buffer location of the ~~at least one~~ incorrectly stored data item to reposition is determined as:

the ~~memory~~ buffer location preceding a first destination ~~memory~~ buffer location, which did not contain any data item; and/or

the ~~memory~~ buffer location following a second destination ~~memory~~ buffer location, which did not contain any data item.

Claim 21 (canceled)

Claim 22 (previously amended): An electronic apparatus for rendering a sequence of interleaved data items, comprising a deinterleaving device for sorting data items according to claim 12.

Claim 23 (previously amended): The apparatus according to claim 22, wherein the apparatus is a mobile radio terminal, a personal digital assistant, a pager, a smartphone, communicator, an electronic organizer, or a multimedia player for rendering digital multimedia files.

Claim 24 (previously amended): The apparatus according to claim 22, wherein the apparatus is a mobile telephone.

Claim 25 (currently amended): A computer program product embodied on a computer readable medium, comprising computer readable instructions to carry out ~~the method according to claim 1~~ a deinterleaving method when run by an electronic device having digital computer processing capabilities, wherein the deinterleaving method is for sorting a sequence of N data items from a first order to a second order, wherein N is an even number, the sequence comprises data items relating to two different sets of data items, and the first order is an interleaving of the two different sets of data items such that the data items from the two different sets of data items are arranged in a list or array alternating one sample relating to each of the two different sets of data items, the method comprising:

withdrawing a pair of data items from said sequence, the pair of data items comprising a first data item and a second data item having respective first and second positions within said sequence;

for each of said withdrawn data items, determining a destination position within said sequence; and

for each of said determined destination positions, determining whether said determined destination position contains any data item, if so replacing the data item of said determined destination position with a respective one of the withdrawn data items, otherwise inserting a respective one of the first data item and the second data item at said determined destination position,

wherein the sequence is indexed from 0 to $N-1$, and wherein the destination position of any incorrectly positioned data item is:

the index of the an original position of the incorrectly positioned data item divided by 2, if the index of the original position of the incorrectly positioned data item is even; or

N divided by 2 and added to the index minus 1 of the original position of the incorrectly positioned data item divided by 2, if the index of the original position of the incorrectly positioned data item is odd.